

Call center

locked

| Problem | Submissions | Leaderboard | Discussions |
|---------|-------------|-------------|-------------|
|---------|-------------|-------------|-------------|

Като тийм лидер в кол център искате да разберете кои ваши подчинени успяват да помогнат на най-много клиенти в един ден. Операторите са k на брой като индексацията започава от 0. Има няколко правила по които се обслужват клиентите:

- Всяко позвъняване на клиент i се определя от две цели положителни числа $t[i]$ и $l[i]$, съответно момента в който човека се е обадил и колко време ще отнема да бъде обслужен. i започва от 0.
- Ако $(i\%k)$ -тия оператор е свободен той ще обслужи клиента.
- Ако $(i\%k)$ -тия оператор е зает ще се пробвате да бъде обслужен от следващия подред свободен оператор, докато не стигнете $k-1$. Ако и той е зает ще се пробвате с оператор 0 и тези след него докато не стигнете $(i\%k)$ -тия отново, ако всички опертори за заети.
- Ако всички оператори са заети в този момент на човека ще бъде автоматично затворено и няма да бъде обслужен.

Забележка: Ако $i\%k = 0$, то се взема последния оператор.

Клиентите се обаждат в строго нарастващ ред, което означава че за всяко i и j , където $i < j$, то $t[i] \leq t[j]$. Това не е задължително вярно за $l[i]$ и $l[j]$.

Намерете индекса на оператора, който е обслужил най-много клиенти. Ако са няколко с максималния брой изкарайте индексите им сортирани в нарастващ ред.

Input Format

На първия ред на стандартния вход ще получите две цели числа N и K , съответно броя на клиентите и броя на операторите. На следващите N на брой реда ще получите по 2 цели положителни числа. Съответно времето, в което ще се обади килиета и колко време ще отнеме да се обслужи

Constraints

$$1 \leq K \leq 10^5 \quad 1 \leq N \leq 10^5 \quad 1 \leq t[i], l[i] \leq 10^9$$

Output Format

На един ред на стандартния изход изведете индексите на операторите, които са обслужили най-много клиенти разделени с интервал сортирани в нарастващ ред.

Sample Input 0

```
5 4
1 3
2 4
3 5
4 4
5 2
```

Sample Output 0

```
0
```

Explanation 0

Има 5 клиента и 4 оператора. Първия клиент се обаждат във време 1 и разговора ще отнеме време 3. Той ще отиде при оператор номер 1 понеже $1 \% 4 = 1$, а първия оператор е този с индекс 0 (4 е броя на операторите, 1 идва от това че това е първото обаждане). Аналогично ще се случи за 2, 3 и 4-тото обаждане които ще отидат при оператори с индекс 1, 2, 3. Обаждане номер 5 ($5 \% 4 = 1$) ще отиде при оператор 1, който е с индекс 0 който и е свободен.

В крайна сметка. Всеки оператори ще са отговорили на едно обаждане с изключение на оператор с индекс 0, който е отговорил на 2. Следователно отговора е 0

Sample Input 1

```
8 4
1 5
2 5
3 3
4 4
5 2
6 2
7 1
8 2
```

Sample Output 1

```
0 2 3
```

Explanation 1

8 клиента, 4 оператора. Обаждане номер 1 отива при оператор 0 и той ще бъде зает до време 6.

Обаждане номер 2 отива при оператор 1 и той ще бъде зает до време 7.

Обаждане номер 3 отива при оператор 2 и той ще бъде зает до време 6.

Обаждане номер 4 отива при оператор 3 и той ще бъде зает до време 8.

Обаждане номер 5 първо отива при оператор с индекс 0, но той е заето до време 6. След това отива до оператор с индекс 1, който също е зает и тн. Няма свободен оператор, така че този разговор ще бъде отхвърлен.

Обаждане номер 6 първо отива при оператор с индекс 1 ($6 \% 4 = 2$, оператор 2 е с индекс 1), но той е зает, следователно ще отиде при следващият. Оператор с индекс 2 е свободен във време 6, така че той ще поеме разговора и ще бъде зает до време 8.

Обаждане номер 7 първо ще отиде при оператор с индекси 2 и 3, но те са заети. Следователно ще отиде при оператор 0, който е свободен. Той ще бъде зает до време 8.

Обаждане номер 8 ще отиде при оператор 3 който е свободен.

Оператор 0, 2, 3 имат по 2 приети обаждания, а оператор 1 има 1. Отговор 0, 2, 3



Submissions: 82

Max Score: 100

Difficulty: Medium

Rate This Challenge:



[More](#)

C++14



```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
```

```
5 #include <algorithm>
6 using namespace std;
7
8
9 int main() {
10     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11     return 0;
12 }
13
```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)

Run Code

Submit Code

[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |